

6 creating and storing a state value for a leaf node of a balanced binary tree, wherein
 7 the leaves of the tree [represents] represent the complete keystream and the
 8 leaf node represents the keystream segment at the location, by a preorder
 9 traversal of the tree from root node to the leaf node wherein a leftward tree
 10 branch transition comprises computing a first non-linear function and a
 11 rightward tree branch transition comprises computing a second non-linear
 12 function;
 13 creating and storing the keystream segment by applying a third function to the state
 14 value of the leaf node.

1 2. (Not amended) A method as recited in Claim 1, further comprising the steps of
 2 creating and storing the balanced binary tree by creating and storing a stack of h
 3 elements wherein the i^{th} element of said stack stores a state datum for the i^{th} node on a
 4 path from a root node of the tree to the leaf node.

1 3. (Not amended) A method as recited in Claim 2, wherein the step of creating
 2 and storing a state value for a leaf node comprises the steps of computing and storing
 3 a state value for the leaf node that is unique with respect to any other state value that
 4 is computed at any other time for any other leaf node of the tree.

1 4. (Canceled.)

12 1 5. (Amended) The method as recited in Claim 1, wherein each leaf node stores $[n] \underline{m}$
 2 bits of state information, wherein $[n] \underline{m}$ is a multiple of [four] twelve.

6. (Amended) The method as recited in Claim 1, further comprising the steps of:
 creating and storing $[3n]$ $m=3n$ bits of state information in each leaf node comprising
 a concatenation of three $[n/2]$ n bit quantities $z|y|x$, wherein n is a multiple
 of four;
 computing the first non-linear function a and the second non-linear function b as the
 composition of a diffusion function d with the nonlinear "confusion"
 functions f and g , wherein $a = f \circ d$ and $b = g \circ d$ and wherein
 $f(z|y|x) = 2z | S(R(S(R(y)))) | L(S(L(S(x))))$
 $g(z|y|x) = 2z+1 | L(S(L(S(y)))) | S(R(S(R(x))))$
 $d(z|y|x) = z | x+y+z | 2x+y+z$
 $c(z|y|x) = x \oplus y$
 wherein integer addition modulo two is denoted as $+$, bitwise exclusive-or is denoted
 as \oplus , and bitwise complementation is denoted as \neg ;
 wherein the R denotes rotation by $n/4$ bits to in a direction of a least significant bit
 and L denotes rotation by $n/4$ bits in a direction of a most significant bit; and
 wherein a nonlinear function S comprises a lookup in a key-dependent substitution
 table.

7. (Amended) The method as recited in Claim 1, wherein the third function
 comprises computing a linear reduction of $[n]$ $2n$ bits of the state value to $[n/2]$ n bits
 thereof.

8. (Not amended) A method as recited in Claim 6, wherein the third function
 comprises computing a bitwise Boolean exclusive OR of x and y .

1 9. (Not amended) A method as recited in Claim 6, further comprising the steps of
2 creating and storing the substitution table S by selecting four invertible functions and
3 storing the four invertible functions in a concatenated form.

1 10. (Not amended) A method as recited in Claim 6, further comprising the steps of
2 computing functions f and g in seven instructions of a central processing unit that can
3 issue two instructions simultaneously, by using five registers to store values of x , y , z ,
4 a temporary variable, and a pointer to the substitution table S .

1 11. (Amended) A method as recited in Claim 6, wherein the substitution table S
2 comprises an array of [randomly selected] key dependent pseudorandom integer
3 values.

13 12. (Amended) A method as recited in Claim 6, wherein the substitution table S
2 comprises an array of 256 [randomly selected] key dependent pseudorandom 32-bit
3 unsigned integer values.

1 13. (Amended) The method as recited in Claim 1, further comprising the steps of
2 creating and storing a key for use by the first non-linear function and the second non -
3 linear function, wherein the key comprises a table of [randomly selected] key
4 dependent pseudorandom values.

1 14. (Amended) The method as recited in Claim 1, further comprising the steps of
2 creating and storing, once and at a time prior to receiving the location value, a key
3 for use by the first non-linear function and the second non-linear function, wherein
4 the key comprises a table of [randomly selected] key dependent pseudorandom
5 values.

1 15. (Not Amended) The method as recited in Claim 1, further comprising the steps
 2 of creating and storing a key in the form of a plurality of pseudo-randomly selected
 3 invertible functions, wherein each of the invertible functions maps an 8-bit portion of
 4 the state value to an 8-bit quantity for use as a substitute portion of the state value.

1 16. (Amended) A method as recited in Claim 1, [wherein the substitution table S
 2 comprises a plurality of sub-tables, and wherein generating the substitution table
 3 comprises (a) setting values of the sub-tables to key-dependent permutations and (b)
 4 setting values of one of the sub-tables to an exclusive OR of itself to the identity
 5 permutation] wherein the pseudo-randomly selected invertible functions are stored in
 6 a plurality of substitution tables, and wherein the plurality of substitution tables are
 7 generated by:
 8 setting each of the plurality of substitution tables equal to the identity function;
 9 for each element of each of the plurality of substitution tables, swapping said element
 10 with another element of such table in a key-dependent manner, and also
 11 performing the same swapping operation on each table that has been
 12 previously been generated.

1 17. (Amended) A method of enciphering a plaintext using at least one keystream
 2 segment at an arbitrary location of a complete keystream, the method comprising the
 3 computer-implemented steps of:
 4 receiving a segment of a plaintext;
 5 receiving a location value that identifies a location of the keystream segment within
 6 the complete keystream;

7 creating and storing a state value for a leaf node of a balanced binary tree, wherein
 8 the leaves of the tree [represents] represent the complete keystream and the
 9 leaf node represents the keystream segment at the location, by a preorder
 10 traversal of the tree from root node to the leaf node wherein a leftward tree
 11 branch transition comprises computing a first non-linear function and a
 12 rightward tree branch transition comprises computing a second non-linear
 13 function;
 14 creating and storing the keystream segment by applying a third function to the state
 15 value of the leaf node;
 16 enciphering the segment of the plaintext by combining the keystream segment with
 17 the segment of the plaintext using a Boolean exclusive OR operation to result
 18 in creating and storing a segment of ciphertext.

1 18. (Amended) A method of encrypting an ordered plurality of packets of a network
 2 communication link using at least one keystream segment at an arbitrary location of a
 3 complete keystream, the method comprising the computer-implemented steps of:
 4 receiving a packet from among the plurality of packets;
 5 determining a location value that represents a relative location of the packet among
 6 the plurality of packets;
 7 creating and storing a state value for a leaf node of a balanced binary tree, wherein
 8 the leaves of the tree [represents] represent the complete keystream and the
 9 leaf node represents a keystream segment at the relative location, by a
 10 preorder traversal of the tree from root node to the leaf node wherein a
 11 leftward tree branch transition comprises computing a first non-linear
 12 function and a rightward tree branch transition comprises computing a second
 13 non-linear function;
 14 creating and storing the keystream segment by applying a third function to the state
 15 value of the leaf node;

enciphering the packet by combining the keystream segment with data of the packet using a Boolean exclusive OR operation to result in creating and storing enciphered packet data.

19. (Amended) A computer-readable medium carrying one or more sequences of instructions for automatically generating a keystream segment of an arbitrary location of a complete keystream of an additive stream cipher, which instructions, when executed by one or more processors, cause the one or more processors to carry out the steps of:

receiving a location value that identifies a location of the keystream segment within the complete keystream;

creating and storing a state value for a leaf node of a balanced binary tree, wherein the leaves of the tree [represents] represent the complete keystream and the leaf node represents the keystream segment at the location, by a preorder traversal of the tree from root node to the leaf node wherein a leftward tree branch transition comprises computing a first non-linear function and a rightward tree branch transition comprises computing a second non-linear function;

creating and storing the keystream segment by applying a third function to the state value of the leaf node.

20. (Amended) An apparatus for automatically generating a keystream segment of an arbitrary location of a complete keystream of an additive stream cipher, comprising: means for receiving a location value that identifies a location of the keystream segment within the complete keystream;

means for creating and storing a state value for a leaf node of a balanced binary tree, wherein the leaves of the tree [represents] represent the complete keystream and the leaf node represents the keystream segment at the location, by a preorder traversal of the tree from root node to the leaf node wherein a leftward tree branch transition comprises computing a first non-linear function and a rightward tree branch transition comprises computing a second non-linear function;

means for creating and storing the keystream segment by applying a third function to the state value of the leaf node.

21. (Amended) An apparatus for automatically generating a keystream segment of an arbitrary location of a complete keystream of an additive stream cipher, comprising: a network interface that is coupled to the data network for receiving one or more packet flows therefrom; a processor; one or more stored sequences of instructions which, when executed by the processor, cause the processor to carry out the steps of: receiving a location value that identifies a location of the keystream segment within the complete keystream; creating and storing a state value for a leaf node of a balanced binary tree, wherein the leaves of the tree [represents] represent the complete keystream and the leaf node represents the keystream segment at the location, by a preorder traversal of the tree from root node to the leaf node wherein a leftward tree branch transition comprises computing a first non-linear function and a rightward tree branch transition comprises computing a second non-linear function; creating and storing the keystream segment by applying a third function to the state value of the leaf node.